The University of Jordan
Authorization Form

I, Maysa' Jamil Farraj , authorize the University of Jordan to supply copies of my Thesis/ Dissertation to libraries or establishments or individuals on request, according to the University of Jordan regulations.

Signature:
Date: 23/5/2011

الجامعة الأردنيـــــة
كلية الدراسات العليا

التاريـــخ: / /

نموذج رقم (١٨)
اقرار والتزام بالمعايير الأخلاقية والأمانة العلمية
وقوانين الجامعة الأردنية وأنظمتها وتعليماتها
لطلبة الماجستير

أنا الطالب: ميساء خليل ابراهيم نزال الرقم الجامعي: (8062138)
تخصص: انظمة المعلومات الكليـــــــة: الملك عبدالله الثاني لتكنولوجيا المعلومات

عنوان الرسالة: ....
Towards a System For Storage and Easy Access
of Cases For a Hotline Help Program
....

اعلن بأنني قد التزمت بقوانين الجامعة الأردنية وأنظمتها وتعليماتها وقراراتها السارية المفعول المتعلقة باعداد رسائل الماجستيرء عندما قمت شخصيا" باعداد رسالتي وذلك بما ينسجم مع الأمانة العلمية وكافة المعايير الأخلاقية المتعارف عليها في كتابة الرسائل العلمية. كما أنني أعلن بأن رسالتي هذه غير منقولة أو مستلة من رسائل أو كتب أو أبحاث أو أي منشورات علمية تم نشرها أو تخزينها في أي وسيلة اعلامية، وتأسيسا" على ما تقدم فانني أتحمل المسؤولية بأنواعها كافة فيما لو تبين غير ذلك بما فيه حق مجلس العمداء في الجامعة الأردنية بالغاء قرار منحي الدرجة العلمية التي حصلت عليها وسحب شهادة التخرج مني بعد صدورها دون أن يكون لي أي حق في التظلم أو الاعتراض أو الطعن بأي صورة كانت في القرار الصادر عن مجلس العمداء بهذا الصدد.

توقيع الطالب:.. التاريخ: ٢٣ / ٥ / ٢٠١١

تعتمد كلية الدراسات العليا
هذه النسخة من الرسالـــة
التوقيع......... التاريخ.٢٣/٥/٢٠١١

TOWARDS A SYSTEM FOR STORAGE AND EASY ACCESS OF CASES FOR A HOTLINE HELP PROGRAM

By

Maysa' Jamil Farraj

Supervisor

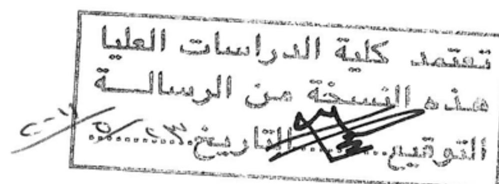
Dr. Nadim Obeid, Prof.

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree in Information Systems

Faculty of Graduate Studies

The University of Jordan

May, 2011



COMMITTEE DECISION

This Thesis/Dissertation (Towards a system for storage and easy access of cases for a hotline help program) was Successfully Defended and Approved on 12/5/2011

<u>**Examination Committee**</u>	<u>**Signature**</u>
Dr. Nadim Obeid, (Supervisor) Prof. of Artificial intelligence	------------------------------
Dr. Amjad Hudaib, (Member) Associate Prof. of Software Engineering	------------------------------
Dr. Thair Hamtini, (Member) Assist. Prof. of E-learning	------------------------------
Dr. Jihad Sa'ady, (Member) Associate Prof. Parallel computing Arab Open University	------------------------------



Dedication

To my parents, brothers and sister for their constant support and love.

To my husband for believing in me.

To my little boy Faris for enlightening my life.

To my friends, managers and fellow workers for their cooperation.

Acknowledgment

I'm thankful to my supervisor Dr.Nadim Obeid for his guidance and support through preparing this thesis. His kindness and understanding made it possible to complete this thesis.

List of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	:	Artificial Intelligence
API	:	Application Programming Interface
DL	:	Description Logic
FAQ	:	Frequently Asked Questions
IR	:	Information Retrieval
JESS	:	Java Expert System Shell
JWU	:	Jordanian Women's Union
NGO	:	Non Governmental Organization
OWL	:	Ontology Web Language
RDF	:	Resource Description Framework
SPSS	:	Statistical Package for the Social Sciences
SQL	:	Structured Query Language
SQWRL	:	Semantic Query-Enhanced Web Rule Language
SWRL	:	Semantic Web Rule Language
W3C	:	World Wide Web Consortium
XML	:	Extensible Markup Language

TOWARDS A SYSTEM FOR STORAGE AND EASY ACCESS OF CASES FOR A HOTLINE HELP PROGRAM

By

Maysa' Jamil Farraj

Supervisor

Dr. Nadim Obeid, Prof.

Abstract

In this research we aim to build a domain ontology model that defines the "violence against women" domain. This model defines the concepts and relationships among them of this domain. The ontology will help in the classification (storage) of cases in order to insure fast and efficient retrieval of cases.

We take the hotline help program as a case study of this domain. The helpline program provides social, psychological and legal services to abused women. In this case study we shall investigate the problems that face this domain.

We shall write several rules in first order logic that manage the retrieval and classification of cases according to specific criteria.

Chapter One:

Introduction

In this thesis we aimed at building a domain ontology application in social services provided for abused women for the hotline program implemented in Jordanian Women's Union in addition to giving a general understanding of ontologies nature and use.

The importance of implementing this ontology for this particular field comes from the large amount of data available within different local institutions that deal with violence against women in Jordan whether governmental or non-governmental.

In the current situation the experts and activists in this field are suffering from the problem of not being able to classify cases and retrieve data effectively for several reasons; first: the data is spread across several sources represented by different organizations that work on violence, second: difficulties in retrieving required information as storage is not effective (whether the data is stored electronically or not).

The usage of ontology will allow experts in this field to model their domain knowledge for more efficient exchange and reuse. By the use of this ontology the program workers and activists will be able to retrieve information and statistics about cases with different relations and variables determined by their needs.

1.1. Problem statement - Hotline Service

1.1.1. What is the Hotline program?

The main objective of the Hotline is to provide a non-prejudiced environment where vulnerable and abused women can speak confidently and freely about their needs

and problems, and receive effective legal, social and psychological counseling provided over the phone or in person.

Once a woman has contacted the Hotline Service, the social worker, along with a lawyer and / or psychologist meets with the women either at the hotline office, or at the women's home. The staff assesses the woman's needs and suggests possible next steps for solution. The woman always has a full range of options presented to her, and should make the final decision for herself. Depending on the needs of the woman, the social worker, psychologist, and lawyer follows the woman's case until a resolution is reached, and conducts regular follow-up to ensure that a durable solution is found.

Possible solutions include, counselling with the household, community and extended family interventions, legal interventions, and, in the most extreme cases, referral to the JWU shelter or other emergency services.

Local authorities are involved from the start in the programme process and in accordance with Jordanian legal conventions. In all of the current Hotline services, local police and social services departments unofficially rely on the JWU Hotline service as a primary support vehicle for vulnerable and abused women.

1.1.2. The problems faced by the Hotline program

The JWU has established the program mentioned earlier in the mid nineties and was one of the pioneer programs in Jordan that addresses and faces this issue.

After all these years of work, what is special about this program is two things, first is the reliability of the information reflecting real life situations for abused women, and second that there is already 7000 different files considered the outcome of almost 15 years of the program. This in fact gives the information available through the records of the hotline program a priority as one of the richest resources for researchers and statistics.

Through working in this organization for almost four years now, I had the chance to address the problems faced by the program. These problems can be summarized in the following:

- No computerized Database of the cases that used the hotline program is available; all the work and documents are still manual. Database of such cases will be of great help to recurring complaint cases and it will help in the follow up.
- Difficult to access relevant documented data and information by hotline staff, researchers and people who are interested in performing appropriate studies on a national level. Even, if data is available, it may prove difficult and a lengthy process to access the relevant files especially when for cases that may be filed under more than one category.

These problems are actually facing the staff and degrading the effectiveness of using these peaces of information for upgrading work or even referencing a specific case

or a specific category. Even when external actors (organizations or individuals) need to benefit from the information available, no accurate or precise information is available. The knowledge and experience of the program is hard to build and maintain. New staff can not benefit from this accumulation of experience.

Creating new programs and expanding current programs is not supported by the needs of abused women and families, as there is no statistical information retrieved depending on the programs records.

Recording the details of the cases follows the special judgment of the responsible social workers. As long as the records are on paper it's harder for the program's managers to check and correct the files of the cases. This resulted in variations in the information included in each record documenting the cases of the program.

It worth mentioning that over 1200 records were lost during transporting the files to the new building of JWU. There was no backup of these records as the system is based on paper recording.

1.1.3. How did the Hotline program tried to solve these problems?

JWU administration has noticed the problems faced by the program – mentioned earlier – and as a result they have tried solving these problems by using the Statistical Package for the Social Sciences (SPSS) (About SPSS Inc, 2011) famous program for statistics, but this system proved it's failure to serve JWU program needs. This failure was because the staff are not qualified enough to lead this

initiative to success. The reasons behind the failure of this initiative can be summarized in the following:

- A technical training on this tool was needed for data entry staff (not social workers. None of the trained staff is still part of JWU and entering the records stopped waiting for new training to be held for new staff.
- The records are supposed to be confidential and restricted to the access of the programs staff. Data entries were not part of the program and that degrades the confidentiality of the program.
- Data entry responsibility of entering the records degrades the quality of the recording, as they miss the professional point of view when recording the files. This should be the responsibility of the social worker if possible.

1.2. Thesis Outline

The rest of this thesis is organized in three chapters as follows:

Chapter two presents relevant review about the fields used in preparing this thesis including ontology, context and context aware and information retrieval.

Chapter three introduces the model developed in addition to results and rules of this model.

Chapter four represents the conclusions and future work.

Chapter Two:

Relevant review

The hotline program provides services to abused women in the Jordanian society. This program has large number of records, and it suffers from the difficulty of classification and storage because of the ambiguity of the cases. To achieve exact classification, appropriate storage and efficient retrieval we needed to build an ontology. In this chapter the concepts of Ontology, Context aware and Information Retrieval were introduced to serve achieving the targeted goals.

2.1Ontology

2.1.1 what is an ontology

There are several definitions for ontology and a hot discussion is still taking place about what ontology is. The following are some of the ontology definitions:

- In philosophy, it means theory of existence. It tries to explain what is being and how the world is configured by introducing a system of critical categories to account things and their relations.

- From Artificial Intelligence (AI) point of view, ontology is defined as "explicit specification of conceptualization" (Gruber, 1993), which is widely accepted in AI community.

  Conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based

system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly. (Gruber, 1993)

- From knowledge-based systems point of view, it is defined as a theory (system) of concepts/vocabulary used as building blocks of an information processing system. In a context of problem solving, ontologies are divided into two types: Task ontology for problem solving process and domain ontology for the domain where the task is performed. (Mizoguchi. Et al., 1995)

- A compositional definition is given by Riichiro MIZOGUCHI as follows: ontology consists of concepts, hierarchical (is-a) organization of them, relations among them (in addition to is-a and part-of) and axioms to formalize the definitions and relations. (Mizoguchi, 2003)

As an example, consider a very simple ontology of part of the animal kingdom (Figure 1). In this ontology, the domain concepts are plants and animals, herbivores and carnivores, antelopes and lions. Carnivores and herbivores are both kinds of animal, so we establish 'is-a' relationships such as 'a carnivore is-a animal' and 'a lion is-a carnivore'. The 'is-a' relation is special because it has transitive properties, allowing us to derive the fact that a lion is an animal, even though this is not stated directly. The 'eats' relation is not transitive, though following it does allow us to analyze the food chain.

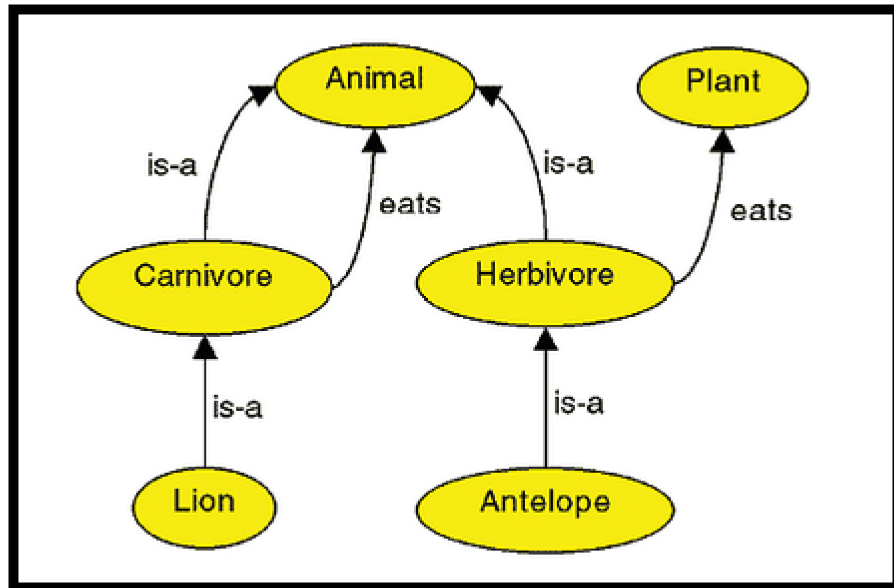


Figure 1: ontology example

2.1.2 Types of ontologies

There are two main classes of Ontologies: the first would be the one that is employed to explicitly capture "static knowledge" about a domain. The second provides a reasoning point of view about the domain knowledge (problem solving knowledge). (Moriya, 2011)

In the first class a distinction between types is made on the basis of the level of generality, as summarized in the table below

Table 1: Ontology class according to level of generality (Moriya, 2011)

1. Domain Ontologies	Designed to Represent knowledge relevant to a certain domain type, e.g. medical, mechanical etc.
2. Generic Ontologies	Can be applied to a variety of domain types. Mereology (Part-Whole theory) Ontologies are applicable to many technical domains. Also called "super theory" and "core technology".
3. Representational Ontologies	These formulate general representation entities without defining what should be represented. The Frame Ontology is a well known example.

For the problem solving knowledge class, two types may be found as shown in table2:

Table 2: Ontology classification according to problem solving type (Moriya, 2011)

1. Task Ontologies	Provide terms specific for particular Tasks.
2. Method Ontologies	Provide terms specific to particular Problem Solving Methods.

2.1.3 What is ontology and what is not?

An ontology is not just a set of terms

While an ontology provides us with a common vocabulary, a vocabulary, that is, a set of terms, itself cannot be said to be an ontology as it is. An ontology needs to consist of an is-a hierarchy of concepts. This is partly because it reveals the proper classification of concepts to show inherent structure of the target world. The reason why an is-a hierarchy of concepts is indispensable is discussed later. Furthermore, there should be a clear distinction between term/word and concepts. The former are "names" of the latter and an ontology is a theory of concepts rather than terms/word. It does not care about how the concept is called. It puts an appropriate name on a concept for making it human understandable. This suggests the synonym is not an ontological issue.

A Heavy-weight ontology is not just a simple hierarchy of concepts

A heavy-weight ontology includes a taxonomy of concepts, but not all taxonomies are a heavy ontology. Let us take an example. Figure 2(a) shows a simple classification of vehicles which includes ground vehicle, car, motor bike, ship and

air craft, etc. It is nothing special. It is not enough to explain or to understand what a vehicle is. In order to clearly understand what a vehicle is, you need to know more concepts such as what function it has, what attribute it has, what machinery it has, how it works in what social environment, etc. Without these concepts, you cannot build a vehicle world. Figure 2(b) shows a hierarchy of such concepts in which the former hierarchy is positioned in one of the categories as type of vehicle. To make such hierarchically organized concepts a heavy-weight ontology, axiomatic definitions of concepts and relations are necessary.

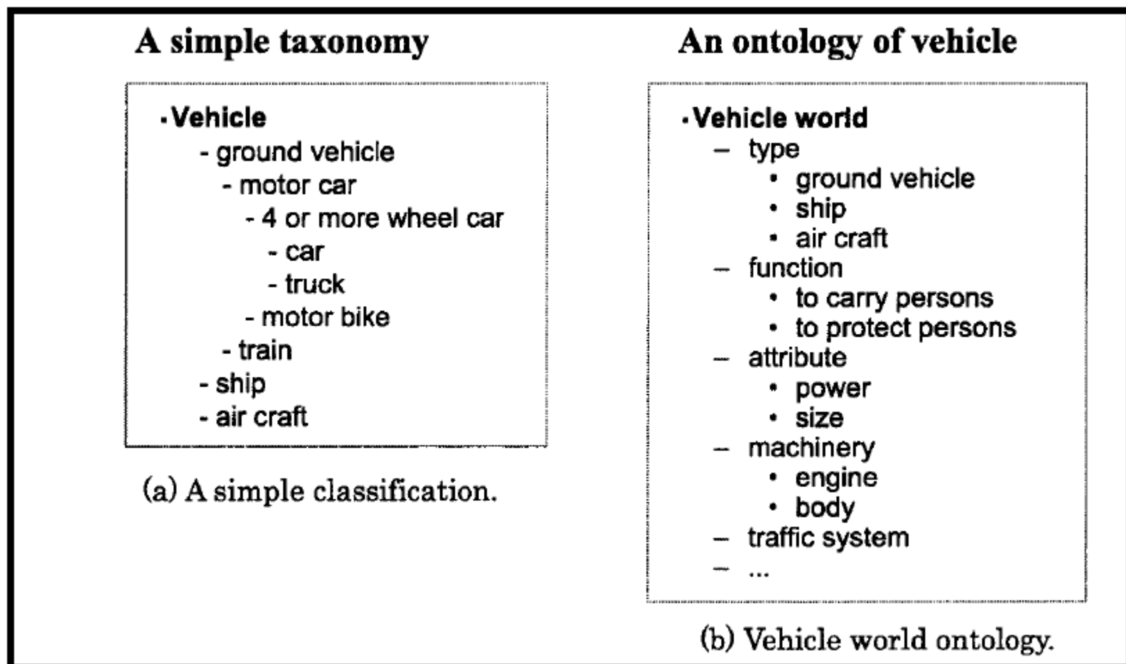


Figure 2: Vehicle Ontology (MIZOGUCHI, 2003)

Ontology is not a knowledge representation

An ontology is neither a semantic network nor a frame. Semantic network and frame are a formalism of the way of knowledge representation. Although an ontology inherently includes is-a hierarchy of concepts, it does not matter whether it is represented by semantic network or frame. In other words, whether a certain thing is

an ontology or not is independent of how it is represented. An ontology provides us with a guideline for modeling the world. To do this, it consists of carefully chosen top-level categories which are reliable enough to explain lower concepts. The ontology problem is thus totally a content issue.

2.1.4 Roles of ontology

Riichiro MIZOGUCHI in his paper "introduction to ontological engineering" has summarized the roles of ontology as follows:

- A common vocabulary.

  The fundamental role of ontology is to give a common vocabulary to be agreed upon by the people involved in the target world description.

- Data structure

  An ontology provides us with a data structure appropriate for information description and exchange.

- Explanation of what is left implicit.

  In all of the human activities, assumptions which are left implicit are found. Ontology is an explication of such implicit knowledge. An explicit representation of such assumptions and conceptualization is more than a simple explication. This explication's contribution to knowledge reuse and sharing is more than expectation considering that the implicitness has been one of the crucial causes of preventing knowledge sharing and reuse.

- Systematization of knowledge

  Well established vocabulary or concepts that describe a reality is a requirement for knowledge systemization. Thus ontology provides a base of systemization of knowledge.

- Meta-model function

  A model is usually built in the computer as an abstraction of the real-world target. And, an ontology provides us with concepts and relations among them which are used as building blocks of the model. Thus, an ontology specifies the models to build by giving guidelines and constraints which should be satisfied.

- Theory of content

  In summary, an ontology provides us with "a theory of content" to enable research results to accumulate like form-oriented research avoiding ad-hoc methodologies which the conventional content-centered activities have been suffering from.

2.1.5 Main components of ontology

- Classes (concepts)

  Represent concepts, which are taken in a broad sense. For instance, in the traveling domain, concepts are: cities, villages, hotels, planes, trains, cars. Classes in ontology are usually organized in taxonomies through which

inheritance mechanisms can be applied. In the frame-based knowledge representation paradigm, meta-classes can also be defined. Meta-classes are classes whose instances are classes. They usually allow for gradations of meaning, since they establish different layers of classes in the ontology where they are defined. (Corch, et al., 2006)

- Relations

  Represent a type of association between concepts of the domain. Ontologies usually contain binary relations. The first argument is known as the domain of the relation, and the second argument is the range. Relations can be instantiated with knowledge from the domain. Binary relations are sometimes used to express concept attributes. Attributes are usually distinguished from relations because their range is a data type, such as string, number, etc., while the range of relations is a concept (Corch, et al., 2006)

- Axioms

  Serve to model sentences that are always true. They are normally used to represent knowledge that cannot be formally defined by the other components. In addition, formal axioms are used to verify the consistency of the ontology itself or the consistency of the knowledge stored in a knowledge base. Formal axioms are very useful for inferring new knowledge. (Corch, et al., 2006)

- Instances

  Individuals (instances) are the basic, "ground level" components of an ontology. The individuals in an ontology may include concrete objects such as people, animals, tables, automobiles, molecules, and planets, as well as abstract individuals such as numbers and words. an ontology need not include any individuals, but one of the general purposes of an ontology is to provide a means of classifying individuals, even if those individuals are not explicitly part of the ontology.

- Attributes

  Objects in an ontology can be described by relating them to other things, typically aspects or parts. These related things are often called attributes, although they may be independent things. Each attribute can be a class or an individual. The kind of object and the kind of attribute determine the kind of relation between them. A relation between an object and an attribute express a fact that is specific to the object to which it is related.

2.1.6 Why we need to build an ontology?

Noy and McGuinness (2001) defines the following reasons for building ontologies:

- To share common understanding of the structure of information among people or software agents.
- To enable reuse of domain knowledge.
- To explicit domain assumptions.

- To separate domain knowledge from the operational knowledge.
- To analyze domain knowledge.

2.1.7 The Development Lifecycle

Although there is some collective experience in developing and using ontologies, there is no field of ontological engineering comparable to knowledge engineering. There are no standardized methodologies for building ontologies. Such a methodology would include a set of stages that occur when building ontologies, guidelines and principles to assist in the different stages, and an ontology life-cycle which indicates the relationships among stages (Uschold and Gruninger, 1996). The most well known ontology construction guidelines were developed by Gruber (Gruber, 1993), to encourage the development of more re-usable ontologies. There has been increased effort in trying to develop a comprehensive ontology methodology (e.g. (Gruninger and Fox, 1995), (Fernandez, et al., 1997), (Uschold and Gruninger, 1996). A survey is given in (Jones, et al., 1998).

In their paper, (Stevens et al, 2000) represented two ontology development lifecycle in figure 3 and figure 4 inspired by the software engineering V-Process Model (Sommerville, 2006).

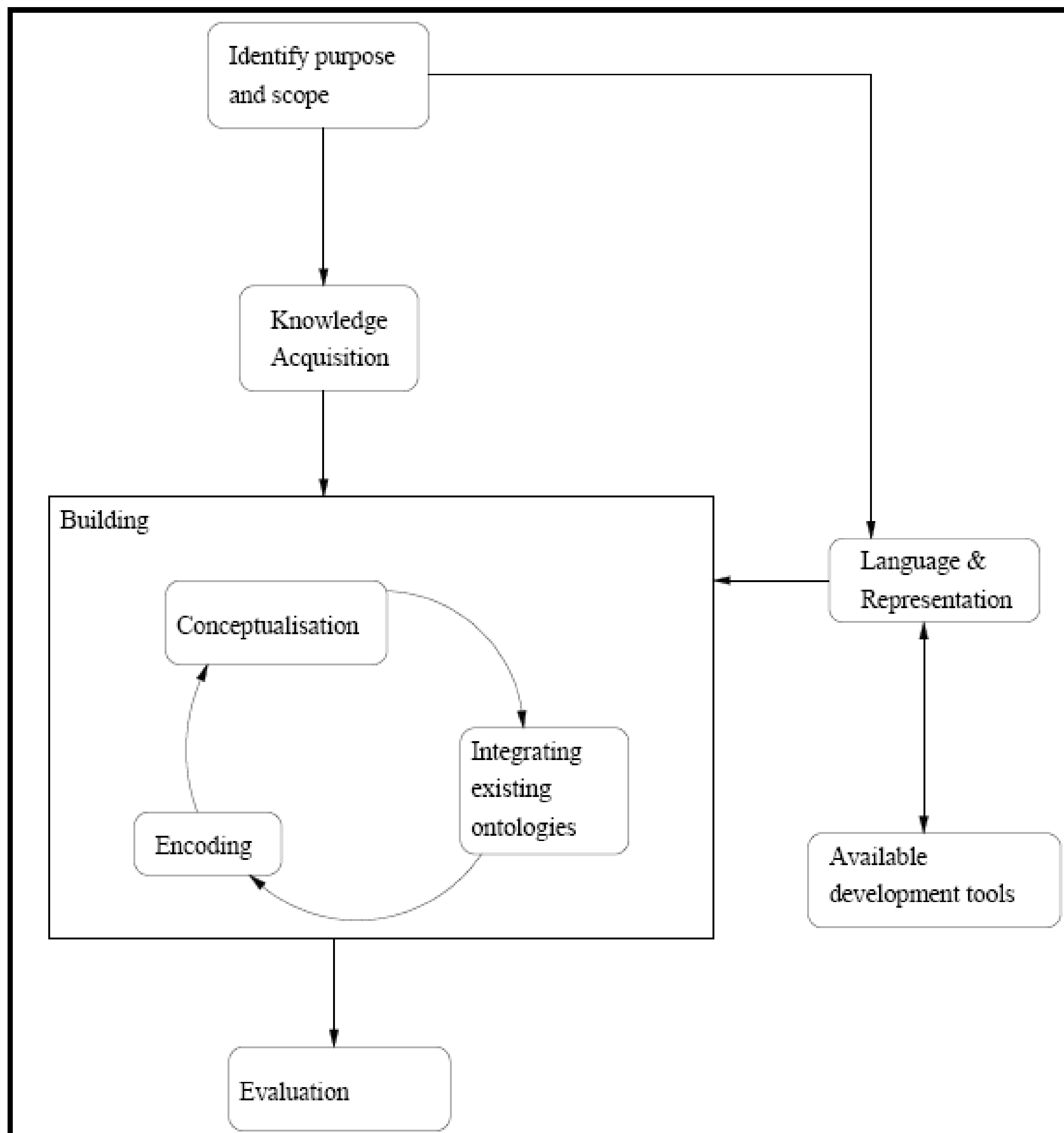


Figure 3: Ontology building life-cycle (Stevens et al, 2000)

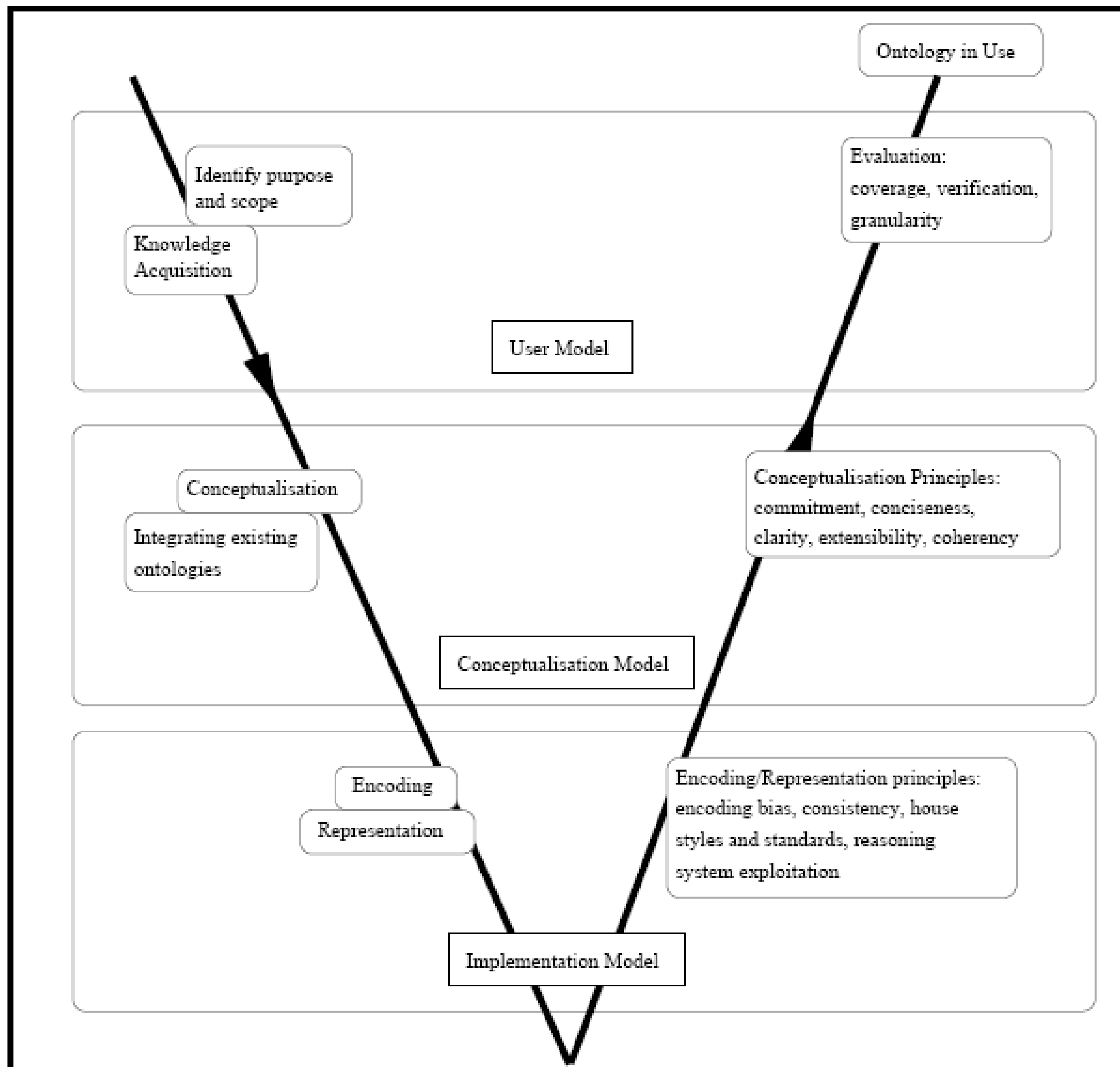


Figure 4: The V-model inspired methodology for building ontologies (Stevens et al, 2000).

According to (Stevens et al, 2000), the stages in the V-process model and life-cycle are:

Identify purpose and scope: developing a requirements specification for the ontology by identifying the intended scope and purpose of the ontology. A well-characterized requirements specification is important to the design, evaluation and re-use of ontology.

Knowledge Acquisition: the process of acquiring domain knowledge from which the ontology will be built. Knowledge holders: Specialist Staff (in our case managers, social workers, psychologists … etc) standard text books and research papers. Informal competency questions formed, the competency questions are used to evaluate the ontological commitments that have been made to see whether the ontology meets the

Requirements. (Uschold and Gruninger, 1996)

Conceptualization: identifying the key concepts that exist in the domain, their properties and the relationships that hold between them; identifying natural language terms to refer to such concepts, relations and attributes; and structuring domain knowledge into explicit conceptual models. This is the process where the concepts and relationships describing the domain are captured. The ontology is usually described using some informal terminology. Gruber (Gruber, 1993) suggests writing lists of the concepts to be contained within the ontology and exploring other ontologies to re-use all or part of their conceptualizations and terminologies. At this stage it is important to bear the results of the first step, that of requirements gathering, in mind.

Integrating: use an existing ontology: a task frequently hindered by the inadequate documentation of existing ontologies. In our study no reusable ontology were found to be used.

Encoding: representing the conceptualization in some formal language, e.g. frames, object models or logic.

Documentation: Documentation is important for defining, more expansively than is possible within the ontology, the exact meaning of terms within the ontology.

Evaluation: determining the appropriateness of an ontology for its intended application. Evaluation is done pragmatically, by assessing the competency of the ontology to satisfy the requirements of its application, including determining the consistency, completeness and conciseness of an ontology.

2.1.8 Ontology Web Language (OWL)

OWL is a family of knowledge representation languages for authoring ontologies. The languages are characterized by formal semantics and RDF/XML-based serializations for the semantic web. OWL is endorsed by the World Wide Web consortium (W3C) and has attracted academic, medical and commercial interest.

2.2 What is context and context-aware?

Humans are quite successful at conveying ideas to each other and reacting appropriately. This is due to many factors: the richness of the language they share, the common understanding of how the world works, and an implicit understanding of everyday situations. When humans talk with humans, they are able to use implicit situational information, or context, to increase the conversational bandwidth.

Unfortunately, this ability to convey ideas does not transfer well to humans interacting with computers. In traditional interactive computing, users have an impoverished mechanism for providing input to computers. By improving the computer's access to context, we increase the richness of communication in human-computer interaction and make it possible to produce more useful computational services. In order to use context effectively, we must understand both what context is and how it can be used. An understanding of context will enable application designers to choose what context to use in their applications. An understanding of how context can be used will help application designers determine what context-aware behaviors to support in their applications.

Context can be seen as a set of information that includes user's activity, location, personal preferences and current status, while the most widely accepted formal definition has been provided by Dey and Abowd (Dey, 1999): ‘‘Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".

Dey and Abowd (Dey, 1999) defines context-aware: "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task".

In their paper Dey and Abowd (Dey, 1999) have chosen a more general definition of context-aware computing. The definitions in the more specific "adapting to context" category require that an application's behavior be modified for it to be considered

context-aware. When a trial to apply these definitions to established context-aware applications, it is found that they do not fit. For example, an application that simply displays the context of the user's environment to the user is not modifying its behavior, but it is context-aware. If less general definitions is used, these applications would not be classified as context-aware. As a result Dey and Abowd chose a more general definition that does not exclude existing context-aware applications.

Context-aware applications look at the who's, where's, when's and what's (that is, what the user is doing) of entities and use this information to determine why the situation is occurring. An application doesn't actually determine why a situation is occurring, but the designer of the application does. The designer uses incoming context to determine why a situation is occurring and uses this to encode some action in the application. For example, in a context-aware tour guide, a user carrying a handheld computer approaches some interesting site resulting in information relevant to the site being displayed on the computer. In this situation, the designer has encoded the understanding that when a user approaches a particular site (the 'incoming context'), it means that the user is interested in the site (the 'why') and the application should display some relevant information (the 'action').

There are three categories that context-aware applications may support:

1. Presentation of information and services to a user.
2. Automatic execution of a service.
3. Tagging of context to information for later retrieval.

2.3 Information Retrieval

Information Retrieval (IR) is the science and technology concerned with the effective and efficient retrieval of information from an information repository for the subsequent use by interested parties. The central problem in IR is the quest to find a set of relevant information resources, amongst a large repository, containing the information sought thereby satisfying an information need usually expressed by a user with a query. The information resources may be objects (items) in any medium, text, image, audio, or, indeed, a mixture of all three.

The basic IR process consists of two steps represented in Figure 5.

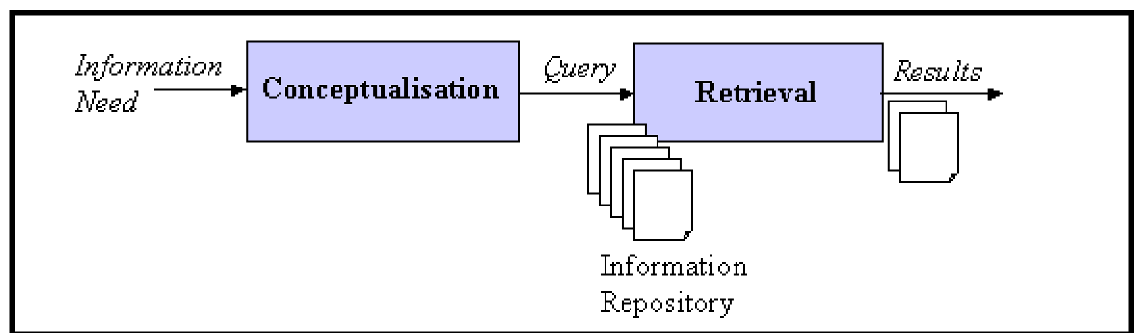


Figure 5: Basic IR process (Stojanovi, 2005)

In the Conceptualization step (see Figure 5), the user tries to represent his information need using the query syntax that underlies that IR process. However, the complexity of an information need is only partially reflected in the query. Indeed, the user usually tries to approximate his need in a query (Saracevic, 1975), by representing only several characteristics of his need, so that the query is usually represented as a set of terms related to some coordinating terms/symbols. Since this

step usually causes ambiguities that appear in a retrieval process, it serves as the main source for the refinement process.

In the Retrieval step (Figure 5) the query is “executed” against the underlying information repository according to a retrieval model, such as Boolean, vector space, probabilistic model (Baeza-Yates, 1999).

Logical IR models were studied to provide a rich and uniform representation of information and its semantics, with the aim to improve retrieval effectiveness. Although the interest in logic of IR researchers may be traced back at least to the early 70's (Cooper, 1971), the first clear statement that IR should be understood in logical terms is due to van Rijsbergen (Van Rijsbergen, 1986). The earliest approaches were directed to the use of classical logic, like Boolean logic.

Chapter Three:

An ontology for a hotline help program

In this thesis we aimed to build an ontology that identifies the reality of a hotline program that provides legal, social, psychological and health care services to women and children, taking into consideration the surrounding implications of our community. In our case, this program is implemented in JWU as mentioned earlier.

This program is suffering from several problems that degrades the level of services provided, and limits the additional benefits that this huge amount of data and information can provide to this field of violence.

The absence of a computerized system that properly categorizes and stores data and information available in this program's records, expose data to be lost, not easily retrieved and no where near categorized.

On the other hand, this amount of information can not be used in any kind of study or acquiring statistical figures that can help on national level improvements or program.

Similar programs are implemented in other organizations (non governmental and semi governmental), and if they are facing the same problems then more information is available that are not used as supposed too. This ontology could be the first step in connecting related services, institutions and programs to build a national level system the support this field. These could include Non-Governmental Organizations (NGOs), courts, family protection unit, ministry of social development.

3.1. Understanding the domain

The first step in developing this ontology is to understand the domain of the hotline service we are intending to define so that the requirements specification is well defined. This helps in the following steps of design, evaluation and effective usage of the ontology. (Stevens, et al., 2000)

The knowledge needed to understand the domain is gathered through several ways that include the following:

- Domain Specialists: these are mainly the program manager. The program manager was one of the founders of the program in 1996; this means that this manager is aware of all the details of this program, the needs of the program, the gaps that are needed to be filled. The manager also is aware of the social context that is related to our society, this is also taken into consideration when explaining the details of this domain.

  The staff (lawyers, social workers and psychological workers) is also aware of the day to day details that enhance the understanding of the domain.

  Meetings were held with these personnel in the stage of understanding the domain.

- Defining the goal of this ontology. This will determine what should be included and what should not. This also specifies the concepts and data to be analyzed. (Brusa, eta al., 2006)

This ontology aims at defining the concepts of violence against women that are dealt with in the hotline program and querying system that helps in 1) retrieving needed information 2) extracting statistical figures that can help addressing problems in this field and proposing solutions.

- Revising paper files and the nature of the data written in these files. This gives a broader idea about the attributes and details that are needed to be included when describing the case.

3.2. Conceptualization

This stage includes identifying the key concepts that exist in the domain, in addition to the relationships between them and the attributes describing them. This identification is the result of the mentioned previously understanding of the domain and determining the requirement specification.

This ontology is considered domain ontology, and in this study it describes the domain of the hotline service provided in JWU. The main objective is to facilitate storage and retrieval of the information related to the cases who receive this service.

In this study, we have worked on defining the concepts of the domain comprehensively accompanied with their relationships (Noy and McGuinness, 2001)

In the following step we have defined the classes, the classes' hierarchy and the properties of the classes.

3.3. Context Model Design

3.3.1. Context Model parts

The model consists of several parts, one of them is considered the core class and named *Person* that represent people related to the universe described by this ontology which is the hotline service. There are six more classes (*LegalAttributes*, *Action*, *ClassViolence*, *RefrralSystem*, *LegalService*, *Assault*) that are separated but related to the core class. These six classes identify the context around the core class *Person* and are related to it through relationships. Context was designed in a way that supports the description of the system and benefits from the advantages that it can provide. Context model main classes are shown below in Figure 6.

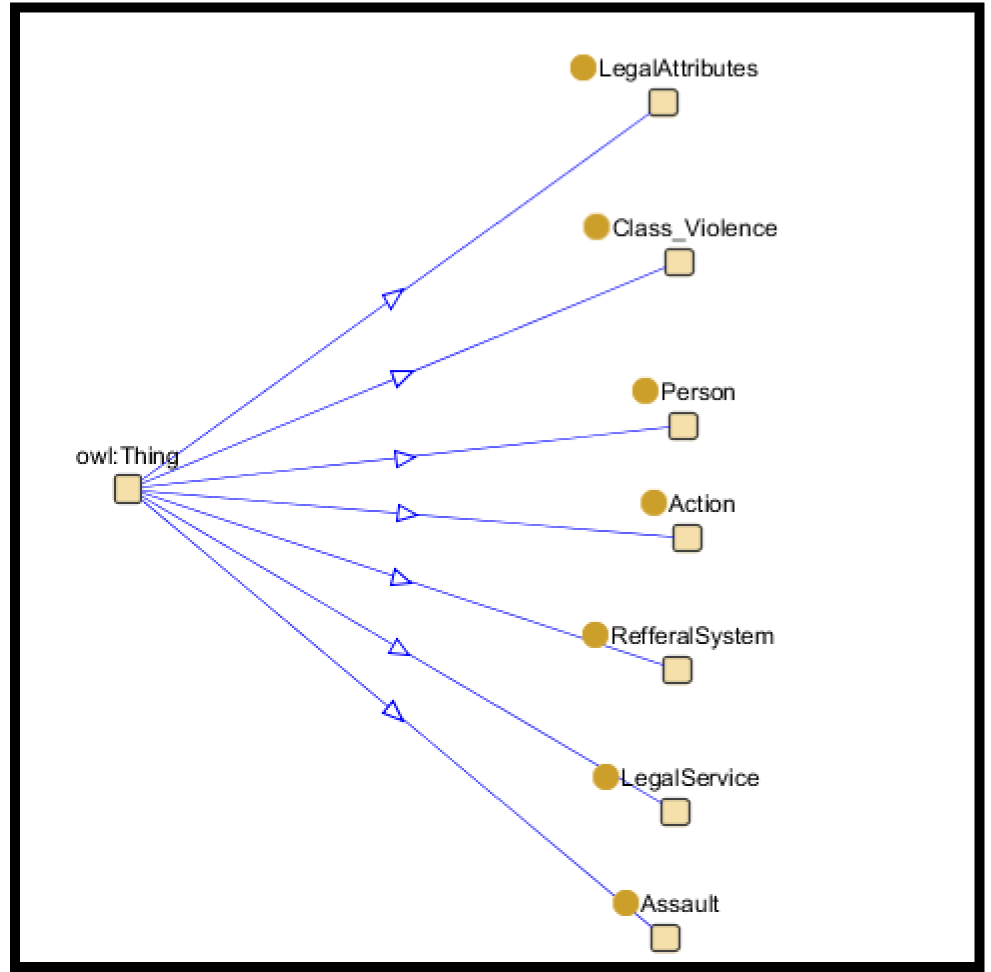


Figure 6: Context Model Classes

Contents of ontology were described earlier in this thesis, which are the same elements that form the basic structure of the proposed model; this can be explained as follows:

1. Object: in the model, concepts are formed into objects and in a hierarchical way, for instance the object *Person* is considered the parent object for *Abusee*, *Abuser* and *Child*.
2. Object Property Relation: are relations that connect concepts together to form a network of concepts, such as concept of *Abusee* has a relation *subjected_to* with a concept of *Assault*.

3. Data Type Relation: it forms the properties of the object and consists of a certain value. The data type of this relation is different according to the value that is needed to be represented. For example the data type of the property *name* which is related to the object *Person* should be string.
4. Axiom: is a set of rules that can describe a concept using logical signs like intersection, union, and others.

In describing the model, OWL-DL language was used in constructing restrictions and defining the object's properties in a logical way. In the other hand, OWL-DL language can be used to achieve consistency of the model and reduce inconsistency problems which could happen during classification processes. Classification processes check consistency and the applicability of the model through assuming several states for each object and its ability to satisfy all the conditions, therefore, if there is any opposition with other states the inconsistency is highlighted, and this means that the existence of this inconsistency would affect the effectiveness of the model in a real process. However, classification processes can be verified according to ability of the object by verifying satisfaction of a certain set of rules. For example, if there is a condition that every person that has a son, he should be a father, therefore, any person satisfies this condition, belongs to the father class. From this point of view, ontology is important in such cases to description logic.

Another important point in DL is the ability of making synonymy in concepts .This would provide several options in describing and giving the one concept several definitions. This can be accomplished through connecting these definitions in a

logical framework as conditions to connect these definitions together and establish relations among them.

3.3.2. The Model Classes

The model consists of one main and core class that is considered the core of the described program, and several classes that describe the domain around this core class. The model consists of the classes described as follows:

1. The core and main class in the proposed model is the class *Person*. This class is the parent class for the classes *Abusee*, *Abuser* and *Child* (Figure 7). These are the main classes in the model as the program serves people mainly, as domestic violence actors.
2. *Action* represents the action taken by the program for the cases (represented by class Person). This class has two sub classes: *FieldVisit* and *FollowUpVisits*. Follow up visits are the visits performed by the abusee to the program, where he/she meets the social worker or the psychologist or the lawyer. Field visits are performed by the social worker accompanied by the psychologist for the place of the victim if he/she can't approach the program for different reasons.
3. A case may need a legal service, which is described by *LegalService* class. This class is related to the class *LegalAttributes* which defines several attributes that describe the legal service provided. These attributes are described by the subclasses: *CourtDecision*, *CustodyDetails*, *DivorceDetails* and *LegalArticles.* Legal articles defines the articles that was used to define the legal articles used in the provided legal service, these legal articles are also classified into three

categories described in the following subclasses: *InsufficientLegislation*, *LackofLegislation* and *SuffecientLegislation*.

4. Every Abusee is subjected to one or more assaults by the abuser that is defined by the class *Assault*. The model suggested different types of assaults depending on the classification mentioned in (Domestic Violence, 2011), and the model has the capability to define more types of assault if it is not listed already in this model.
5. Some of the cases of the program are referred to it by other stakeholders, governmental and non-governmental organizations. These stakeholders are defined through the class *RefferalSystem*.
6. Each case is classified through a rule to fit in one of the classes that defines the types of violence. These classes are subclasses of *Class_Violence* and these subclasses are: *Physical*, *Emotional*, *Sexual*, *Verbal* and *Economic* (Domestic Violence, 2011).

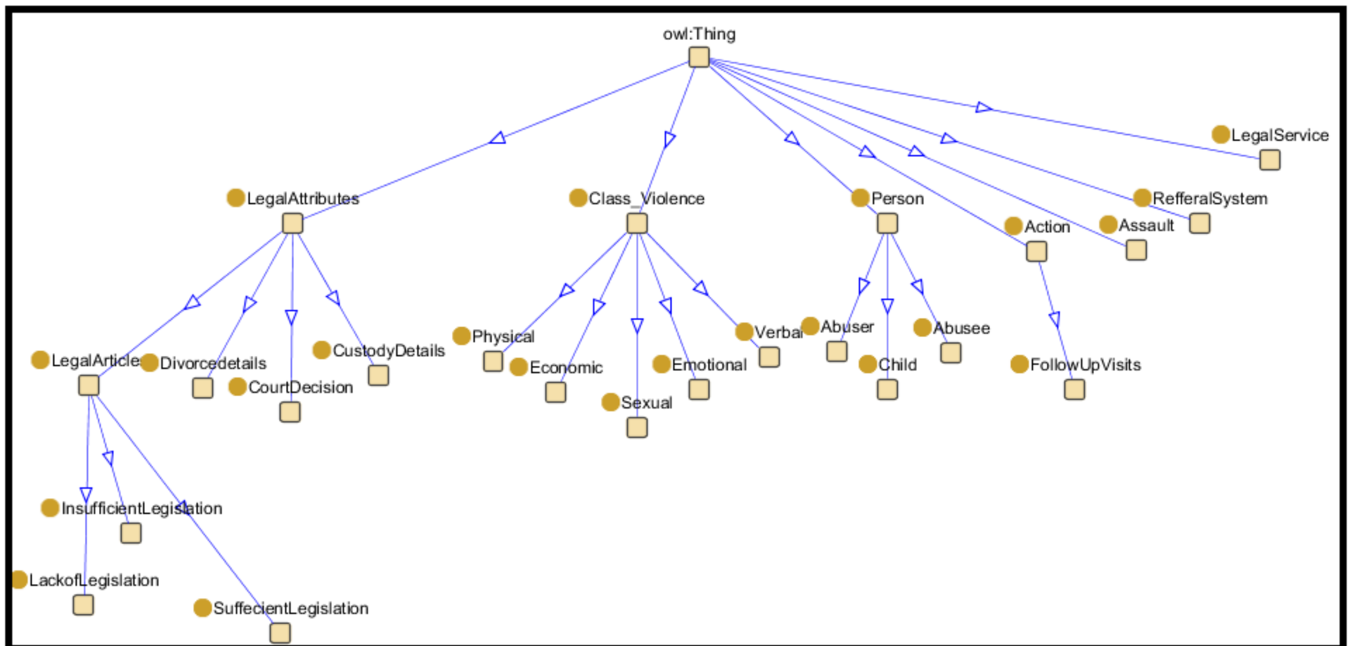


Figure 7: The Model Classes – expanded

3.3.3. Objects' property relationships

Object properties link an individual to an individual. Properties may be created using the ‘Object Properties’ tab shown in Figure 8.

There are several types of object property relationships:

1. Functional properties: If a property is functional, for a given individual, there can be at most one individual that is related to the individual via this property.
2. Inverse functional properties: If a property is inverse functional then it means that the inverse property is functional. For a given individual, there can be at most one individual related to that individual via the property.
3. Transitive properties: If a property is transitive, and the property relates individual a to individual b, and also individual b to individual c, then we can infer that individual a is related to individual c via property P.

   Symmetric properties: If a property P is symmetric, and the property relates individual a to individual b then individual b is also related to individual a via property P.

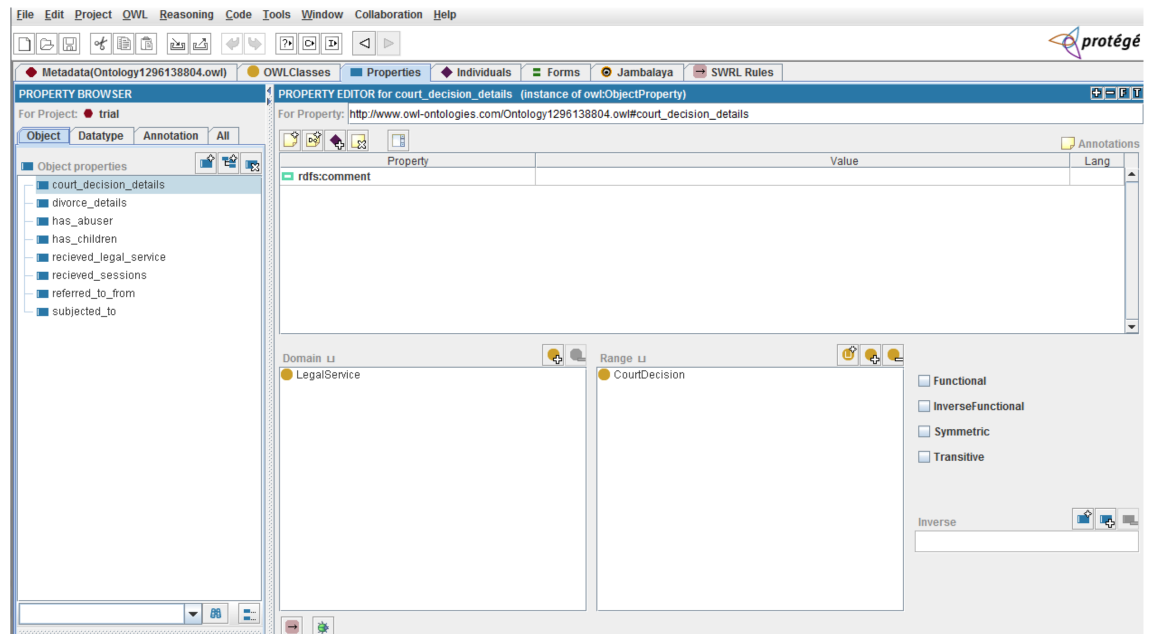


Figure 8: Object Property Tab.

Property relationships within class *Person*:

- Classes *Child*, *Abuser* and *Abusee* are subclasses of the class *Person*.
- Classes *Abusee* and *Child* are related through the relation *has_children*. Where the relationship is represented as

  "Abusee ----has_children----→ Child"

- Classes *Abusee* and *Abuser* are related through the relation *has_Abuser*. Where the relationship is represented as

  "Abusee ----has_Abuser----→ Abuser"

  These relationships are represented in figure 9 below.

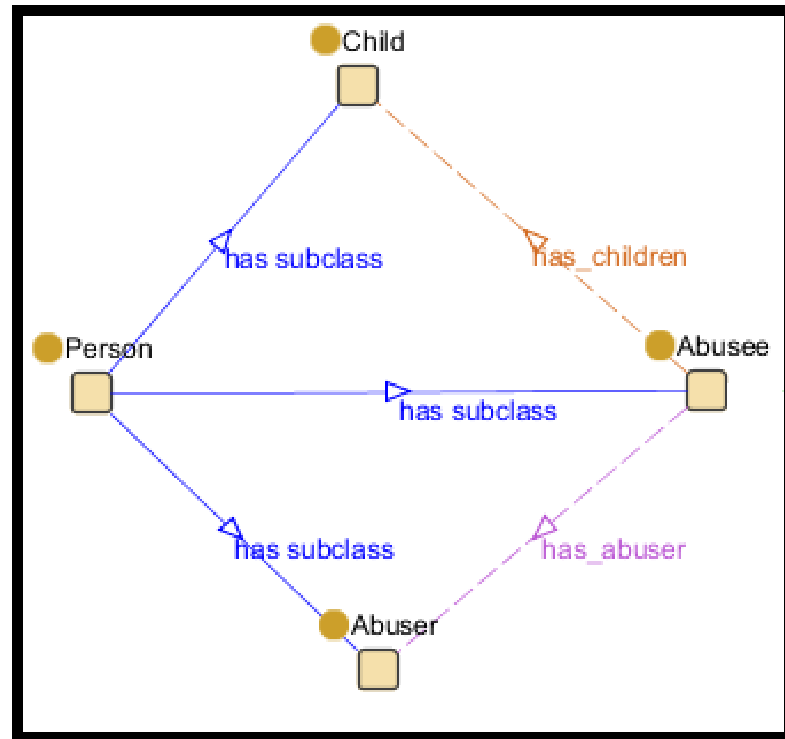


Figure 9: Property relationships within class *Person*

Property relationships related to class *Abusee*:

- Classes *Abusee* and *LegalService* are related through the relation *received_legal_service*. Where the relationship is represented as

  "Abusee ---- received_legal_service ----→ LegalService"

  This relationship defines the legal service that the abusee receives from the program when applicable.

- Classes *Abusee* and *Assault* are related through the relation *subjected_to*. Where the relationship is represented as

  "Abusee ---- subjected_to ----→ Assault"

  This relationship assigns to each abusee the assault(s) that he/she is subjected to. The model defines a set of assaults that covers several aspects of what an abusee can be subjected to. The model also gives the user the ability to define a new type of assault if not already defined by the model.

- Classes *Abusee* and *RefferlSystem* are related through the relation *reffered_to_from*. Where the relationship is represented as

  "Abusee ---- *reffered_to_from* ----→ *ReferralSystem"*

  This relationship defines the organization where the abusee is referred from where applicable. This information helps in the process of following up the case and managing relationships with other local and international stakeholders.

- Classes *Abusee* and *FollowUpVisits* are related through the relation *received_sessions*. Where the relationship is represented as

  "Abusee ---- *received_sessions* ----→ *FollowUpVisits"*

  This relationship defines the social or psychological sessions that the program performs with the case. This piece of information is considered as a record of what are the steps that have been done with each case.

These relationships are represented in figure 10 below.

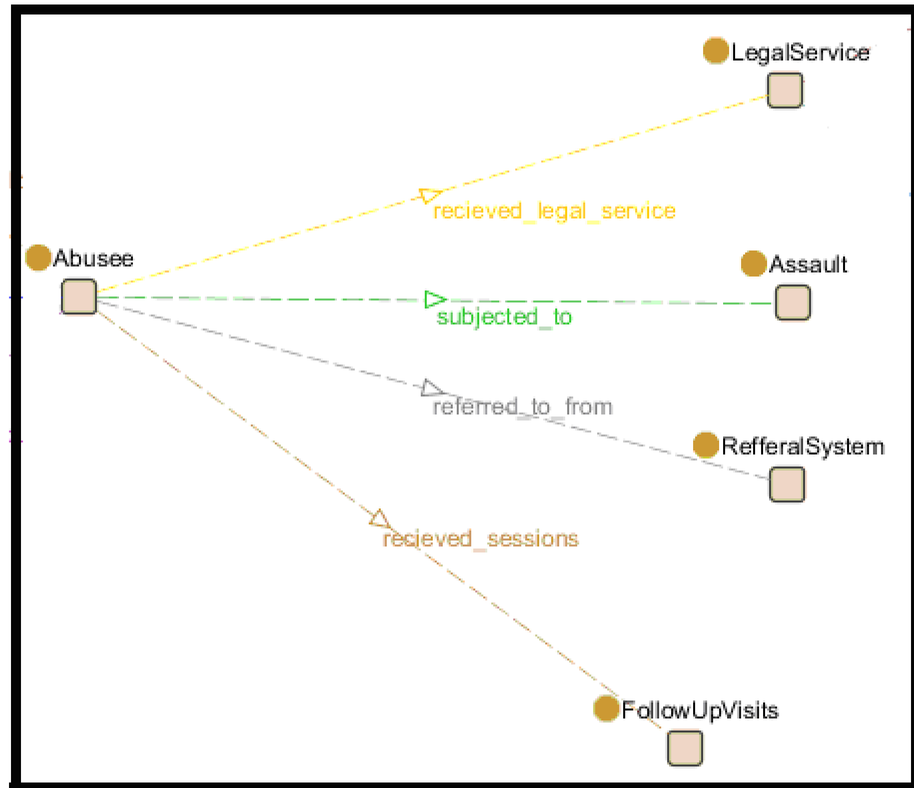


Figure 10: Property relationships related to class *Abusee*

Property relationships related to class *LegalService*:

- Classes *LegalService* and *CourtDecision* are related through the relation *court_decision_details*. Where the relationship is represented as

  "LegalService ---- court_decision_details ----→ CourtDecision"

  This relationship defines the court decision that is taken as a result of the legal service provided for each case when applicable.

- Classes *LegalService* and *Divorcedetails* are related through the relation *divorce_details*. Where the relationship is represented as

  "LegalService ---- divorce_details ----→ Divorcedetails"

  This relationship defines the details of a divorce court decision that is taken as a result of the legal service provided for each case when applicable. This specificity

is given to the divorce decisions because it contains several details and it is a frequent situation that faces the program.

These relationships are represented in figure 11 below.

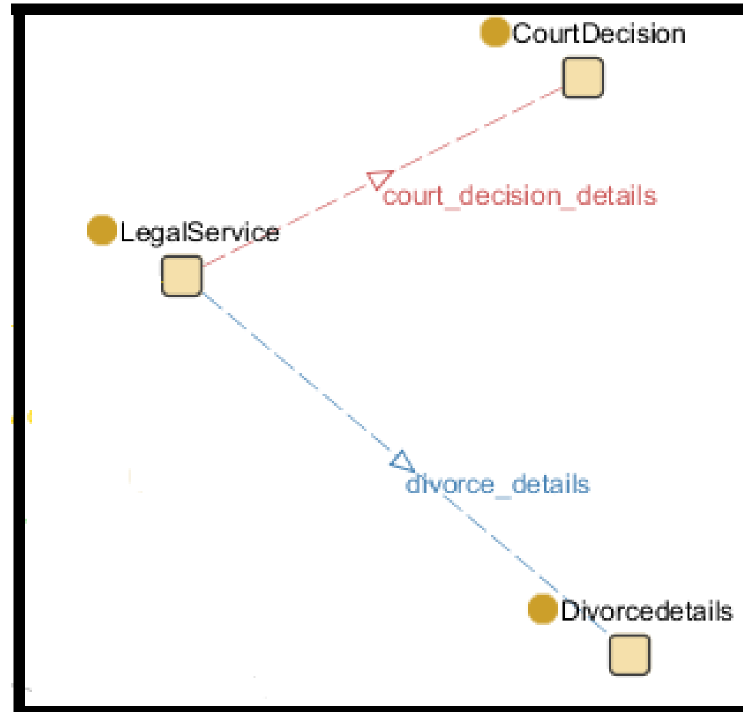


Figure 11: Property relationships related to class *LegalService*

3.3.4. Data Type Relation

In this model several data types defines the properties of each class. Datatype properties link an individual to an XML Schema Datatype value or an rdf literal. In other words, they describe relationships between an individual and data values. Datatype properties can be created using the 'Datatype Properties' tab shown in Figure 12 (Horridge, et al., 2007).

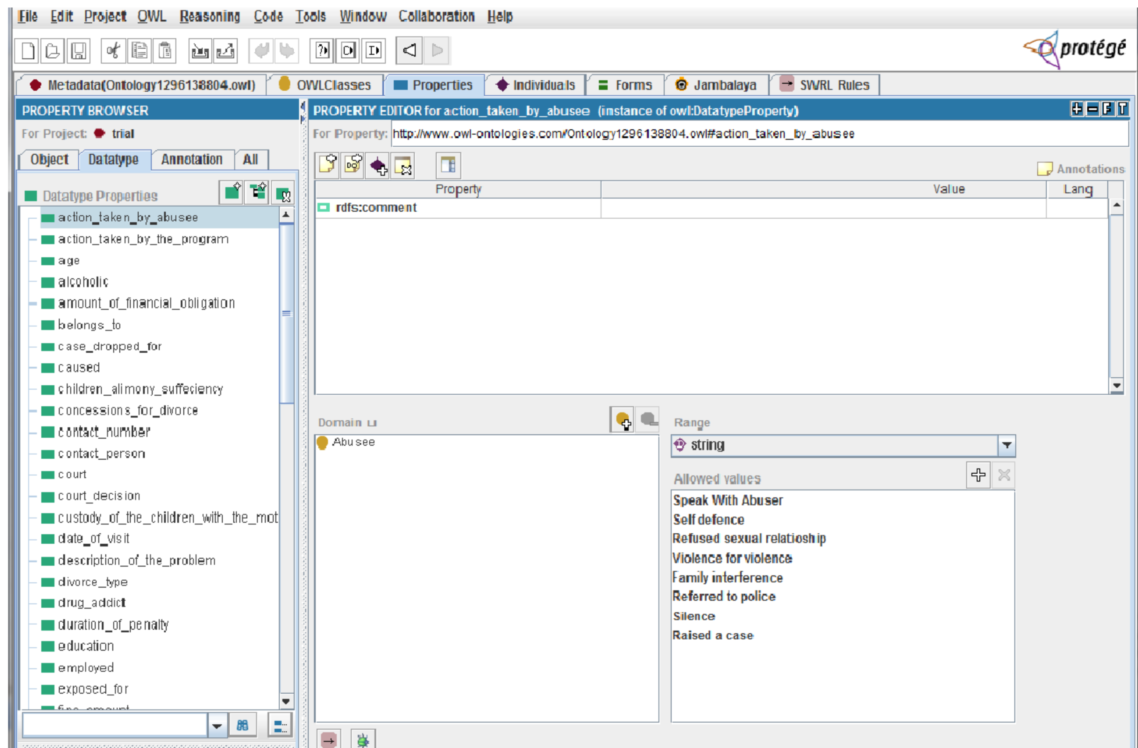


Figure 12: Data properties tab

In this section we will write down a detailed description of the data type properties of each class in the model:

1. Class *Person* has the following data type properties:

- serial number : an integer unique for each person
- age and it's an Integer
- description_of_the_problem: represents description of the problem as described by this person and it's a String.
- education it's a string and takes a value of the following: Illiterate, Elementary, Secondary, Diploma, Bachelor Degree and Post graduate.
- health_status it's a string and takes a value of the following: Healthy, Handicapped and Chronic disease.
- marital_status it's a string and takes a value of the following: Single, Married, Divorced, Separated, Engaged and Widow.

- name and it's a string.
- nationality and it's a string and takes a value of the following (these values where determined according to the recommendation of the program managers depending on the percentage of each nationality): Jordanian, Moroccan, Egyptian, Palestinian, Other Arabic nationalities and foreign.
- sex is a string and takes a value of the following: male and female.

2. Class *Abusee* has the following data type properties:
   - It inherited all the data type properties of the class Person as the class Abusee is a subclass of the class Person.
   - action_taken_by_abusee represents the action/s that the abusee has tried as a possible solution for his/her problem before asking for the program's help. It is a string and it takes a value of the following: Speak with the abuser, Self defense, Refused sexual relationship, Violence for violence, Family interference, Referred to police, Silence, Raised a case.
   - action_taken_by_the_program represents the action/s that the program has performed as part of the process performed with each abuse. It is a string that takes a value of the following: Listening, Legal consultation, Judicial service, Health care service, Financial support, Psychological support, Social help, Shelter, Help in education and empowerment, Offer job, Vocational training, Physiatrist, Repairing the relationship.
   - belongs_to is a string that describe the nature of community that the abusee belongs to, it takes a value of the following: City, Village, Camp.
   - date_of_visit is of date data type and gives the first time the abusee visited the program.

- employed is of Boolean data type, and indicates if the abusee is employed or not.
- Knew_about_the_program_from is of string data type. It indicates how the abusee knew about the program. It takes one of the following values: Newspaper, TV, Radio, Advertisement, Friends, Brochure, Doctor, Neighbors, Other.
- income is of integer data type, determines the value of the abusee's income.
- ownership_of_residence is of string data type. Takes one of two values: Owned or Rented. This gives an indicator of the financial situation and stability of the abusee.
- resides_in is of string data type and it takes one of the following values: Villa, House, Room, Hostel. This property also gives an indicator the financial situation and stability of the abusee.
- resides_in_neighborhood is of type string and takes one of two values: wealthy neighborhood and popular neighborhood. This property also gives an indicator the financial situation and stability of the abusee.
- responsible_lawyer is of string data type. It is an input property that holds the name of the responsible lawyer in the program that follow up a particular abusee.
- responsible_psychologist is of string data type. It is an input property that holds the name of the responsible psychologist in the program that follow up a particular abusee.
- responsible_social_worker is of string data type. It is an input property that holds the name of the responsible social worker in the program that follow up a particular abusee.

- social_worker_description_of_the_problem is of string data type. It is an input property that contains a detailed description of the abusee situation. It also contains the professional analysis of the case.
- social_worker_recommendation is of string data type. It is an input property that contains recommendations from the social worker of the practical actions to be taken when dealing with the abuse.
- telephone is of type integer. An input property that contains the contact telephone number of the abusee.

3. Class *Abuse*r has the following data type properties:
   - It inherited all the data type properties of the class Person as the class Abuser is a subclass of the class Person.
   - belongs_to is a string that describe the nature of community that the abuser belongs to, it takes a value of the following: City, Village, Camp.
   - employed is of Boolean data type, and indicates if the abuser is employed or not.
   - income is of integer data type, determines the value of the abuser's income.
   - alcoholic is of type Boolean. This property indicates if the abuser has alcohol addiction problem or not. Usually this type of problems usually increases the possibilities of abuse.
   - drug_addict is of type Boolean. This property indicates if the abuser has drug addiction problem or not. Usually this type of problems usually increases the possibilities of abuse.

- relation_to_the_abusee is of type string. It identifies the relation of the abuser to the abuse, and it takes one of the following values: Husband, Brother, Father, Son, Mother, Ex-husband.

4. Class *Assault* has the following data type properties:
   - caused is of type string. It indicates the effects of the assault an abusee is subjected to and it takes a value of the following: Psychological effects, Social effects, Bruses, Broken bones, Permanent disability, Temporary disability.
   - exposed_for is of type string. It determines an assault or more that an abusee is exposed to and it takes a value of the following: Punching, 'Pushing, Pulling', Slapping, Striking with an object, Kicking, Kneeing, Strangling, Drowning, Sleep deprivation, Exposure to heat or radiation, Exposure to electric shock, Placing in stress position, Cutting, Withholding food or medication, Biting, stabbing, poisoning, Rape, Marital rape, Forced sex, Sexual harassment, Threatening physical harm, Forcing isolation from family, Destruction of property, Rejecting, Degrading, Consistent criticizing, Manipulation of words, Humiliation, Making others feel unwanted and unloved, Accusing, Insulting, Name calling, Yelling, Acquisition of salary, Preventing from work, Forced Begging.
   - performed_in is of type string and indicates the place where an assault is performed. It takes one of the following values: Work, Home, University, School, Street, Market, Any where.
   - performed_during is of type string and indicates the time during which an assault is performed. It takes one of the following values: Day, Night, Weekends, Vacations, Any time.

5. Class *FollowUpVisits* has the following data type properties:

   - date_of_visit is of Date data type and represents the date of each follow up visit performed for an abusee.
   - social_worker_minutes_of_visit is of string data type. It is an input property that includes the minutes of a follow up visit; it is filled by the responsible social worker that performs the session.

6. Class *CourtDecision* has the following data type properties:

   - amount_of_financial_obligation is of Integer data type. It is an input property that includes the financial obligation that a court decides for the program's case – if applicable-.
   - court is of String data type. It indicates the type of the court that is concerned with the type of the service needed by the program's case. This property takes a value of the following: Ecclesiastical Court, Shar'ieh court, Order court – if applicable - .
   - court_decision is of type String and contains the court decision description – if applicable – and it is an input property that is filled by the responsible lawyer.
   - duration_of_penalty is of type Integer contains the duration of penalty if the court decision contained a jailing decision.
   - fine_amount is of type integer. Contains the fine amount if the court decision contained a fine penalty. It is an input property.
   - the_decision_contained is of type String. Contains an indicator of the court decision. It takes one of the following values: Custody, Alimony, Divorce, Financial obligation, Compensation, Fine, Jail.

7. Class *Refferalsystem* has the following data type properties:
   - contact_number is of type Integer. It is an input property that contains a contact number of the organization or institution that referred the case to the program.
   - Contact_person is of type String. It is an input property that contains a contact person name in the organization or institution that referred the case to the program.
   - name_of_referring_organization is of type String. It is an input property that contains the name of the organization or institution that referred the case to the program.
   - nature_of_organization is of type String. This indicates the nature of the referring organization, and it takes one of the following organizations: Government, NGO, Private Sector.

8. Class *LegalService* has the following data type properties:
   - was_the_case_dropped is of type Boolean. Indicates if an abusee has dropped a judicial case or not after raising it.
   - case_dropped_for is of type String. It indicates the reasons for dropping a legal case –if applicable- it takes a value of the following: Financial reasons, No supporting family members, Threats, Fear of loosing children, Reconciliation.
   - implementation_procedures is of type String. It gives a description of the complexity of the legal procedures and it takes a value of the following: Simple, Complicated, Lengthy, Short. This indicated by the judgment of the case's responsible lawyer.

- is_a_case_raised is of type Boolean. It indicates if an abusee has raised a case or not.
- provided_service_details is of type String. It is an input property that includes a description written by the responsible lawyer about the details of the provided service concerning the legal part of the whole services provided by the program.
- reasons_for_not_raising_a_case is of type String. Defines the reasons behind refusing raising a case by the abusee when applicable. It takes a value of the following: No will, Lack of money, Children best interest, No supporting family member, Long Litigation.
- service_needed_is is of type String. It defines the services needed by the abusee as requested by the abusee and recommended by the responsible lawyer. It takes a value of the following: Legal consultation, Judicial case, Follow up of legal procedures.
- targeted_law is of type String. it is an input property where the responsible lawyer defines the targeted law article that supports the best interest of the abusee.

9. Class *DivorceDetails* has the following data type properties:

- children_alimony_suffeciency is of type Boolean. In the case of divorce, this property indicates if the decided alimony for children is enough or not.
- wife_alimony_suffeciency is of type Boolean. In the case of divorce, this property indicates if the decided alimony for the ex-wife is enough or not.

- concessions_for_divorce is of type string. It defines the concessions that follow of a divorce decision. It takes one of the following values المؤخر، توابع المهر، النفقة الزوجية، نفقة الأطفال، نفقة التعليم، بدل حضانة و سكن.
- divorce_type is of type String. it defines the type of divorce decision and it takes a value of the following: رجعي، بائن بينونة كبرى، بائن بينونة كبرى، طلاق مقابل ابراء، خلع، مخالعة.
- implementation_procedures is of type String. It gives a description of the complexity of the legal procedures and it takes a value of the following: Simple, Complicated, Lengthy, Short. This indicated by the judgment of the case's responsible lawyer.

10. Class *CustodyDetails* has the following data type properties:
    - custody_of_the_children_with_the_mother is of type Boolean. It indicates if the custody is with the mother or not in the case of divorced families.
    - the_mother's_parents_attitude_toward_her_children_custody is of type string. it gives an indicator of the obstacles that may face a mother in getting her children custody and it takes a value of the following: Supporting, Refusing.
    - implementation_procedures is of type String. It gives a description of the complexity of the legal procedures and it takes a value of the following: Simple, Complicated, Lengthy, Short. This indicated by the judgment of the case's responsible lawyer.
    - was_the_custody_decision is of type String. it defines how the custody decision was taken and it takes one of two values: Judicial, Consensual.

11. Class *LackofLegislation* has the following data type properties:

- Unexisting_law_article is of type String. is an input property that includes the un-existing law article that supports the best interest of the abusee.
- suggested_law_legislation is of type String. it is an input property that includes a legal suggestion of the legal article that can be added. This helps when collecting information about law reforms when needed.

12. Class *InsuffecientLegislation* has the following data type properties:
    - suggested_law_legislation is of type String. it is an input property that includes a legal suggestion of the legal article that can be added. This helps when collecting information about law reforms when needed.

3.3.5. Connection to Database

Protégé tool gives us the ability of storing records using and retrieving them using the SWRL tab using the first order logic rules. We were not able to test the storage capabilities of protégé as we only had a testing sample of 20 cases; as a result we needed to build a data base to store cases.

The previous model represented by its classes and relationships can be matched with a database easily to store records. Each class from the previous classes represents a table in the database. Each property relationship represents the relationship between the concerned tables. Data type relationships represent the columns of the tables. The unique serial number is the key of the different tables. The is-a relationship represents the inheritance relationships.

The rules of the ontology perform pre processing on the cases, so that retrieval is done intelligently taking benefit of the intelligent storage. Searching through a

logical tree is more time efficient as search is not performed over all the cases but only over the cases that matches the criteria defined by the query rules.

3.4. Implementing tool used: Protégé 3.4.4

3.4.1. Definition and Features

Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. (protégé, 2010)

Protégé has attractive advantages (Simov, 2004) that include and not limited to:

- Open source and easy to install

  Protégé is an open source platform for ontology creation and visual ontology manipulation. The installation process is easy.

- Plug-in Architecture

  On of the most valuable advantages of Protégé is its Plug-in architecture. Using Protégé in that direction provides a platform for the development of additional functionalities specific for the project. The developed modules will interoperate with the available ones in Protégé. The most convenient way for our purpose is the development of “tab plug-ins”. The tabs may be switched on and off depending on the user’s needs.

- Manipulation of huge ontologies

  A number of ontology tools were tested (Simov, 2004) Protégé showed the most satisfactory results on loading different ontologies.

- Protégé Maintenance

Protégé has a satisfactory maintenance environment. There are forums, developer email list in case of a problem, Frequently Asked Questions (FAQ) list, Plug-in library that contains a lot of plug-ins separated by their functionality, tutorials for using Protégé and developing modules for Protégé, updates and new versions are also available.

- Interoperability with Jena Model and Pellet Reasoner.

3.4.2. Implementation

For the advantages mentioned earlier, we used the implementation of this ontology we used protégé 3.4.4. We chose to use it because it's an open source tool that had a forum discussing and answering problems that faces users in addition to the fact that it is extensible and provides a plug-and-play environment that makes it a flexible base for application development (Knublauch, et al., 2005).

In the implementation, the Context Interpreter uses the protégé to design ontology model, manipulate with contextual information and inference contextual knowledge, where we build rule based using SWRL a Java plug-in that can be used to create and manipulate Semantic web Rules, SWRL tab developed to build applications rule to support Semantic Web scenarios, Beyond the support of RDF, DAML+OIL (Van, et al., 2001) and OWL languages, and it configures with jess reasoner agent also offers components to build rule Inference engines.

Java Expert System Shell (JESS) agent (JESS, 2011) and protégé are tools to build context domain, where they provide many utilities and different techniques for

evaluating ontology model, where jess provide agent to reasoning knowledge rule. On the other hand, there are many plug-in tabs and Java Application programming interface (API) that are configuring with protégé, this provides many services that support creating and representing ontologies such as jena (Carroll, et al., 2004) API, TGVIZtab (Alani,2003) .

3.5. Ontology Querying

3.5.1. What is SWRL?

The Semantic Web Rule Language (SWRL) is an expressive OWL-based rule language. SWRL allows users to write rules that can be expressed in terms of OWL concepts to provide more powerful deductive reasoning capabilities than OWL alone. Semantically, SWRL is built on the same description logic foundation as OWL and provides similar strong formal guarantees when performing inference. (SWRL language FAQ, 2011)

SWRL is based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language. SWRL allows users to write Hornlike rules expressed in terms of OWL concepts to reason about OWL individuals. The rules can be used to infer new knowledge from existing OWL knowledge bases. (O'Connor, et al., 2005)

SWRL rules are written as antecedent-consequent pairs. In SWRL terminology, the antecedent is referred to as the rule body and the consequent is referred to as the

head. The head and body consist of a conjunction of one or more atoms. At present, SWRL does not support more complex logical combinations of atoms.

SWRL rules reason about individuals, primarily in terms of OWL classes and properties. For example, a SWRL rule expressing that a person with a male sibling has a brother requires capturing the concepts of 'person', 'male', 'sibling' and 'brother' in OWL. Intuitively, the concept of person and male can be captured using an OWL class called Person with a subclass Man; the sibling and brother relationships can be expressed using OWL properties hasSibling and hasBrother, which are attached to Person. The rule in SWRL would then be:

Person (?x1) ^ hasSibling(?x1,?x2) ^ Man(?x2) → hasBrother(?x1,?x2)

The previous rule is considered a general rule but SWRL can refer explicitly to OWL individual as the following rule that infers that a particular individual Ahmad has a brother:

Person(Ahmad) ^ hasSibling(Ahmad,?x2) ^ Man(?x2) →

hasBrother(Ahmad,?x2)

SWRL also supports the common same-as and different-from concepts. For example, the SWRL sameAs atom can determine if two OWL individuals Ahmad and Hammoudeh are the same individual:

sameAs(Ahmad, Hammoudeh)

Similarly, the differentFrom atom can be used to express that two OWL individuals are not the same.

SWRL also supports a range of built-in predicates, which greatly expand its expressive power. SWRL built-ins are predicates that accept several arguments. They are described in detail in the SWRL Built-in Specification. The simplest built-ins are comparison operations. For example, the greaterThan built-in determines if an individual has an older brother.

hasBrother(?x1,?x2) ^ hasAge(?x1,?age1) ^ hasAge(?x2,?age2) ^ swrlb:greaterThan(?age2,?age1) → hasOlderBrother(?x1,?x2)

The SWRL editor also supports more complex mathematical built-ins. For example, the following rule determines if an individual has a brother who is exactly 10 years older:

hasBrother(?x1,?x2) ^ hasAge(?x1,?age1) ^ hasAge(?x2,?age2) ^ swrlb:subtract(10,?age2,?age1) → hasDecadeOlderBrother(?x1,?x2)

A list of the SWRL built-ins is defined on The DARPA Agent Markup Language Homepage (SWRL Built-Ins, 2011).

3.5.2. What is SQWRL?

SQWRL (Semantic Query-Enhanced Web Rule Language; pronounced squirrel) is a SWRL-based language for querying OWL ontologies. It provides SQL-like operations to retrieve knowledge from OWL.

SQWRL is defined using a library of SWRL built-ins that effectively builds a query language on top of SWRL. These built-ins are defined in the SQWRL ontology. It has the default namespace prefix SQWRL. A copy of this ontology can be found the standard Protege-OWL repositories, or can be imported through the 'Import Ontology' option in Protege-OWL's Metadata tab. The Jess rule engine is currently required to run SQWRL queries.

SQWRL queries can operate in conjunction with SWRL and can thus be used to retrieve knowledge inferred by SWRL rules.

Assume, for example, the following rule that classifies persons older than 17 as adults:

Person(?p) ^ hasAge(?p, ?age) ^ swrlb:greaterThan(?age, 17) → Adult(?p)

A query to list all the adults in an ontology can be written:

Adult(?p) → sqwrl:select(?p)

SQWRL queries operate on known individuals in the currently loaded OWL ontology. It is very important to note that SQWRL provides no way of accessing the information it accumulates from within a rule so query results cannot be written back to the ontology. There is no way, for example, to insert the result of a computed aggregate count back into the ontology. Such a mechanism could invalidate OWL's open world assumption and lead to nonmonotonicity.

While SQWRL's operators are implemented as SWRL built-ins, they do not operate like standard built-ins. Unlike most built-ins, they do not evaluate their arguments and return true if the arguments satisfy some predicate. Instead, they always return true and act as accumulators and build up table-based data structures outside of an ontology. However, they are side-effect free in terms of this ontology - they do not perform any ontology modifications. Crucially, they do not violate SWRL's semantics.

Unlike OWL and SWRL, SQWRL adopts the unique name assumption when querying.

3.5.3. The suggested model rules and queries

Taking into consideration the needs of the program as described by the program managers and staff, the following queries are examples of the needed information by experts and activists working in this field to be extracted.

Rule 1:

Abusee(?x) ^ Responsible_social_worker(?x, "ميسر اسماعيل")

→ sqwrl:select(?x)

This rule retrieves simple information that is a list of the cases that was the responsibility of a specific social worker –Myassar in this example-. This peace of information is needed by the program locally for the purposes of evaluation and follow up. More details could be added to such a query.

Rule 2:

Abusee(?x) ^ age(?x, ?ag) → sqwrl:avg(?ag)

This rule retrieves statistical information about the average age of the cases received by the program. This peace of information could be used as part of larger information containing different variables. This rule used a built-in function to calculate the average of the age.

Rule 3:

Abusee(?x) ^ has_children(?x, ?y) °

sqwrl:makeBag(?b, ?y) ^ sqwrl:makeSet(?bb, ?x) °

sqwrl:size(?size_b, ?b) ^ sqwrl:size(?size_bb, ?bb) ^

swrlm:eval(?av, " size_b/size_bb", ?size_b, ?size_bb) →

sqwrl:select(?av)

This rule calculates the average of the number of children of the abused women who approach the program. We needed to use three built ins (makeBag, makeSet, size) to store and retrieve the number of children. We also used the built in (eval) to get the value of the calculation that results in the average value of the children number.

Rule 4:

Abusee(?x) ^ has_abuser(?x, ?y) ˚

sqwrl:makeSet(?s, ?y) ˚

sqwrl:size(?physical_abused_alcoholic, ?s) ^ Abusee(?z) ^

has_abuser(?z, ?yy) ^

sqwrl:makeSet(?ss, ?yy) ^ sqwrl:size(?physical_abused, ?ss) ^

alcoholic(?y, true) ^ Physical(?x) ^ Physical(?z) ^

swrlm:eval(?percentage, ", physical_abused_alcoholic/physical_abused",

?physical_abused_alcoholic, ?physical_abused) →

sqwrl:select(?physical_abused_alcoholic, ?physical_abused,

?percentage)

This rule calculates the percentage of the alcoholic abusers among the physical abuse cases. This information – and other similar – gives an indicator of the nature of abusers among specific type of abuse. More specific actors could be added to calculate such indicators for research purposes.

Rule 5:

- **Abusee(?x) ^ subjected_to(?x, Pushing) → Physical(?x)**
- **Abusee(?x) ^ subjected_to(?x, Forced_begging) → Economic(?x)**
- **Abusee(?x) ^ subjected_to(?x, Insulting) → Verbal(?x)**
- **Abusee(?x) ^ subjected_to(?x, Degrading) → Emotional(?x)**
- **Abusee(?x) ^ subjected_to(?x, Rape) → Sexual(?x)**

These are part of a set of rules that classifies the cases that approaches the program to five types of abuse (physical, emotional, economical, verbal and sexual) according to what they where exposed to. This classification helps in retrieving information about the cases that approach the program, for instance this classification was used in rule 4.

Rule 6:

Abusee(?x) ^ Social_worker_recomedation(?x, ?rec) ^ Physical(?x) → sqwrl:select(?rec)

This rule retrieves the social worker recommendations when a case is physically abused. This piece of information could help in evaluating the type of recommendation given to the cases and if these recommendations comply with the program's goals and strategies or not.

Rule 7:

Abuser(?x) ˚

sqwrl:makeSet(?s, ?x) ˚

sqwrl:size(?abuser, ?s) ^ Abuser(?z) ^ alcoholic(?z, true) ^

sqwrl:makeSet(?ss, ?z) ^ sqwrl:size(?alcohol, ?ss) ^

swrlm:eval(?percentage, " alcohol/abuser", ?alcohol, ?abuser) →

sqwrl:select(?alcohol, ?abuser, ?percentage)

This rule calculates the percentage of alcoholic abusers among abusers records of the program. This calculation could be copied to different and/or several factors that give an indicator of the factors that increase the possibility of abuse against women and/or children.

3.5.4. Context Awareness Scenarios

Scenario 1:

Fatema is an Abusee. When she approached the program with bruises and broken bones then the social worker first impression was that she was physically abused.

The rule that reflects this scenario is:

Abusee(?x) ^ subjected_to(?x, ?y) ^ caused(?y, "Bruses") ^

caused(?y, "Broken boes") →

Physical(?x)

Scenario 2:

Fatema is an abusee when she approached the program she was scared, hallucinating, 20 years old, living with the extended family and her cloths were torn. The social worker first impression was that she was sexually abused.

The following rule reflects this scenario:

Abusee(?x) ^ Social_worker_description_of_the_problem(?x, ?y) ^ swrlb:contains(?y, "scared") ^ swrlb:contains(?y, "hallucinat")^ swrlb:contains(?y, "torn cloths") ^ age(?x, ?ag) ^ swrlb:lessThan(?ag, 35) ^ swrlb:greaterThan(?ag, 15) → Sexual(?x)

Through the previous scenarios context awareness importance was expressed through classification of the cases based on the context. The class of violence a cased is exposed for is determined through several variables.

There are several combinations of these variables that can effect the classification of the cases, this is because social work field is human oriented, and human nature is very complicated and every detail may change the whole understanding and accordingly the class of a case. Many situations are considered sensitive and may need quick response according to the sensitivity of the case, for instance a life threatened case may need to be sheltered immediately to protect her life.

All these classifications may help the program staff taking the needed and appropriate decisions as defined by the corresponding rule.

3.5.5. Results and discussion

The previous rules were tested on a sample of 20 real cases. The rules expressed examples of the needed information to be retrieved taking into consideration different criteria. These rules express the potential and variety of rules and information that can be retrieved.

The rules worked well, and retrieved the needed information. These rules only give an indication of the potentials of this ontology, and the wide variety of rules that can be built upon the requests and needs of the program staff and managers.

Chapter Four:

Conclusions and future work

4.1. Conclusions

In this thesis we built a domain ontology that helped in the classification process of the cases of a help line program. This classification leads to an intelligent storage and more efficient retrieval. We used "protégé 3.4.4" tool to build the ontology. This tool gave us the ability to query the cases using first order logic rules; retrieval is more effective because cases are retrieved logically. This made information retrieval more time efficient especially when considering the large number of records. We defined several rules that retrieves and classifies cases depending on several criteria. These rules where tested over a sample of 20 cases taken from the records of the hotline program implemented in Jordanian women's union.

During the building process of our ontology we defined the concepts of this domain after understanding it through personal interviews and reviewing paper work of the program, and then we implemented it using Protégé 3.4.4. We defined the classes of the domain, and the relationships among them but we haven't defined the axioms yet because understanding the complexed domain has consumed the biggest margin of the working time.

4.2. Future work

There are several aspects where this work can be improved and enhanced. Our future work should include the following:

- Complete the ontology by defining the axioms and consulting more experts in the fields of psychology, law and social science.
- Building a natural language processing application that can read the documents and use it as an input for our application.

- Finding a way to apply similar ontology application in different institutions, to have a national record in the field of violence against women.

REFRENCES

About SPSS Inc. (2011) retrieved on May 13th 2011 from **http://www.spss.com/corpinfo/history.htm**

Alani, Harith (2003), **TGVizTab: An Ontology Visualisation Extension for Protégé**, Workshop on Visualization Information in Knowledge Engineering , Sanibel Island, Florida, USA.

Baeza-Yates, Rcardo (1999), Berthier Ribeiro-Neto, **Modern Information Retrieval**, Addison-Wesley- Longman Publishing.

Brusa, Graciela. and Caliusco, Ma. Laura.(2006), **A Process for Building a Domain Ontology: an Experience in Developing a Government Budgetary Ontology.**

Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., and Wilkinson, K.(2004), **Jena: implementing the semantic web recommendations**. In Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers &Amp; Posters (New York, NY, USA, May 19 - 21, 2004), WWW Alt. '04. ACM, New York, NY, 74

Cooper, W.S. (1971), **A definition of relevance for Information Retrieval**, in Information Storage and Retrieval, Volume 7, pp. 19-37.

Corch, Oscar. Fernández-López, Mariano. and Gómez-Pérez, Asunción. (2006), **Ontological Engineering: Principles, Methods, Tools and Languages,** Springer 1-48

Dey, K. and Abowd, G.D. (1999), **Towards a Better Understanding of Context and Context- Awareness**, Technical Report GIT-GVU-99-22, GVU Center, Georgia Institute of Technology.

Domestic Violence (2011), extracted from **http://en.wikipedia.org/wiki/Domestic_violence** on April 13th 2011

Fernandez, M. Gomez-Perez, A. and Juristo, N. (1997), **METHODONTOLOGY: From Ontological Art to Ontological Engineering**, In Spring Symposium Series, pages 33.40, Menlow Park, California, AAAI Press.

Gruber, Thomas R. (1993), **Toward Principles for the Design of Ontologies Used for Knowledge Sharing**, Stanford Knowledge Systems Laboratory

Gruninger, M. and Fox, M. S. (1995), **Methodology for the Design and Evaluation of Ontologies**, In *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*. Published on-line by Ceur Publication **http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/.**

Horridge Matthew. Jupp, Simon. Moulton, Georgina. Rector, Alan. Stevens, Robert. and Wroe, Chris. (2007), **A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools Edition 1.1,** The University Of Manchester

JESS 2011, retrieved on April 13th 2011 from **http://herzberg.ca.sandia.gov/jess/**

Jones, D.M. Bench-Capon, T.J.M. and Visser, P.R.S. (1998), **Methodologies for Ontology Development**. In J. Cuena, editor, Proc. ITi and KNOWS Conference of the 15th IFIP World Computer Congress, pages 62.75, London, UK. Chapman and Hall Ltd

Knublauch, H. Horridge, M. Musen, M. Rector, A. Stevens, R. Drummond, N. Lord, P. Noy, N. Seidenberg, J. and Wang, H. (2005), The Protégé OWL Experience, Workshop on **OWL: Experiences and Directions**, Fourth International Semantic Web Conference (ISWC2005), Galway, Ireland.

Mizoguchi, Riichiro. Ikeda, Mitsuru. Seta, Kazuhisa. And Kakusho, Osamu. (1995), **Task Ontology for Reuse of Problem Solving Knowledge**, in Proc. ofKB&KS95, Enshede, The Netherland.

Mizoguchi, Riichiro (2003), **Tutorial on Ontological Engineering part 1: Introduction to Ontological Engineering,** The Institute of Scientific and Industrial Research, Osaka University.

Moriya, Netzer. (2011), **the specification of a shared conceptualization**, Retrieved April 13th 2011 from **http://www.iqlue.com/Ontology.jsp**

Noy, Natalya F. and McGuinness, Deborah L. (2001), **Ontology Development 101: A Guide to Creating Your First Ontology**, Stanford University

O'Connor, M. J. Knublauch, H. Tu, S. W. and Musen, M. A. (2005), **Writing Rules for the Semantic Web Using SWRL and Jess**, Protégé with Rules Workshop, held with 8th International Protégé Conference, Madrid, Spain.

Protégé (2011), retrieved on April 13th 2011 from **http://protege.stanford.edu**

Saracevic, Tefko. (1975), **Relevance: A Review of and a framework for the thinking on the notion in information science**, in Journal of the American Society for Information Science, Volume 26, Number 6, pp. 321-343.

Simov, Alexander. (2004), **A semantic-based knowledge flow system for the European home textiles industry**

Sommerville, Ian. (2006), Software Engineering 8th edition.

Stevens, Robert. Goble, Carole A. and Bechhofer, Sean. (2000), **Ontology-based Knowledge Representation for Bioinformatics**, University of Manchester

Stojanovi, Nenad. (2005), Ontology**-based Information Retrieval: Methods and Tools for Cooperative Query Answering**, Fridericiana the University of Karlsruhe.

SWRL Built-Ins (2011) retrieved on April 13th 2011 from **http://www.daml.org/2004/04/swrl/builtins.html**

SWRL language FAQ (2011) retrieved from **http://protege.cim3.net/cgi-bin/wiki.pl?SWRLLanguageFAQ**

SQWRL (2011) retrieved on April 13th 2011 from **http://protege.cim3.net/cgi-bin/wiki.pl?SQWRL**

Uschold, Mike. and Gruninger, Michael. (1996), **Ontologies: Principles, Methods and Applications,** Knowledge Engineering Review.

Horrocks,I.,Van Harmelen, F., Patel-Schneider, P. F., Lee,T.B., Decker, D.C.M.S., Fensel,D., Fikes,R., Hayes,P., Hendler,P., McGuinness,O.L.D., Stein,L.A.(2001), **DAML+OIL, Agent Markup Language Committee**. Retrieved at December 10 2010 **http://www.daml.org/2001/03/daml oil.daml .**

van Rijsbergen, C. J. (1986), **A new theoretical framework for information retrieval**, In Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'86), Pisa, Italy, pp. 194–200.

Wang, James Farha Ali, Z. and Srimani, Pradip K. (2010), **An efficient method to measure the semantic similarity of ontologies**, International Journal of Pervasive Computing and Communication, Vol. 6 No. 1.

Get Information from Ontology using SQWRL in Protégé (2010) retrieved at April 13th 2010 at **http://yuhanaresearch.files.wordpress.com/2007/11/get-information-from-ontology-using-sqwrl.pdf** written by Yuhana (Nov, 22nd 2007)

نحو نظام لتخزين و استرجاع حالات برنامج خط الإرشاد النفسي و الاجتماعي و القانوني

إعداد

ميساء جميل فراج

المشرف

الأستاذ الدكتور نديم عبيد

ملخص

تهدف هذه الدراسة الى بناء انطولوجي لتعريف مجال "العنف ضد المرأة " حيث يتم تعريف المفاهيم و العلاقات فيما بينها . ستساعد الانطولوجي في تصنيف و تخزين الحالات لتحقيق استرجاع سريع و فعال للمعلومات.

تم تنفيذ الدراسة على حالة دراسية تتمثل في برنامج خط الارشاد الذي يقدم خدمات النفسية و القانونية و الاجتماعية للنساء المعنفات . من خلال هذه الحالة الدراسية تم استقصاء و تحديد المشكلات التي يعاني منها هذا المجال.

تم تعريف و كتابة قواعد لاسترجاع و تصنيف الحالات حسب متغيرات مختلفة.